

KU LEUVEN**STADIUS**Center for Dynamical Systems,
Signal Processing and Data Analytics

Citation/Reference	Alexander Bertrand and Marc Moonen (2015), Distributed adaptive generalized eigenvector estimation of a sensor signal covariance matrix pair in a fully-connected sensor network Signal Processing, vol. 106, pp. 209-214, Jan. 2015.
Archived version	Author manuscript: the content is identical to the content of the published paper, but without the final typesetting by the publisher
Published version	http://dx.doi.org/10.1016/j.sigpro.2014.07.022
Journal homepage	http://www.journals.elsevier.com/signal-processing
Author contact	alexander.bertrand@esat.kuleuven.be + 32 (0)16 321899
IR	https://lirias.kuleuven.be/handle/123456789/458433

(article begins on next page)

Distributed adaptive generalized eigenvector estimation of a sensor signal covariance matrix pair in a fully-connected sensor network

Alexander Bertrand and Marc Moonen

KU Leuven, Dept. of Electrical Engineering (ESAT)
Stadius Center for Dynamical Systems, Signal Processing and Data Analytics
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium

E-mail: alexander.bertrand@esat.kuleuven.be
marc.moonen@esat.kuleuven.be

Phone: +32 16 321899, Fax: +32 16 321970

Abstract—The generalized eigenvalue decomposition (GEVD) of a pair of matrices generalizes the concept of the eigenvalue decomposition (EVD) of a single matrix. It is a widely-used tool in signal processing applications, in particular in a context of spatial filtering and subspace estimation. In this paper, we describe a distributed adaptive algorithm to estimate generalized eigenvectors (GEVCs) of a pair of sensor signal covariance matrices in a fully-connected wireless sensor network. The algorithm computes these GEVCs in an iterative fashion without explicitly constructing the full network-wide covariance matrices. Instead, the nodes only exchange compressed sensor signal observations, providing a significant reduction in per-node communication and computational cost compared to the scenario where all the raw sensor signal observations are collected and processed in a fusion center.

Index Terms—Wireless sensor networks, distributed estimation, generalized eigenvalue problem

I. INTRODUCTION

The so-called generalized eigenvalue decomposition (GEVD) of a pair of matrices generalizes the eigenvalue decomposition (EVD) of a single matrix [1]. The GEVD is often used for subspace estimation or noise reduction, as it reveals a linear transformation that maximizes the signal-to-noise ratio (SNR) [2]–[4]. Furthermore, it is also used for direction-of-arrival estimation [5], for blind source separation based on second-order-statistics [6], and to extract feature vectors with good discriminative properties, e.g., in brain-computer-interfaces [7].

In this paper, we consider the GEVD problem in a wireless sensor network (WSN), where we aim to estimate the Q generalized eigenvectors (GEVCs) corresponding to the Q largest or smallest generalized eigenvalues (GEVLs) of a pair of a-priori unknown sensor signal covariance matrices,

which capture the cross-correlation between all sensor signal pairs. In principle, one could transmit all the raw sensor signal observations to a fusion center (FC) to construct both covariance matrices, and then compute their GEVD. However, such data centralization requires a significant communication bandwidth and significant computational power at the FC. In this paper, we propose a distributed algorithm to avoid this data centralization.

This distributed GEVC estimation problem is a generalization of the problem statements in [8]–[10], which address the distributed estimation of (non-generalized) eigenvectors (EVCs) of a single network-wide sensor signal covariance matrix. The algorithms in [8], [9] are operated in a WSN with an ad hoc topology and are based on a distributed realization of the power method [8] or Oja's learning rule [9], both in combination with nested consensus averaging iterations. In [10], the nodes locally compute the EVCs of compressed covariance matrices, which are estimated from compressed sensor signal observations. As opposed to [8], [9], the algorithm in [10] is operated in a WSN with a fully-connected or tree topology, which avoids the need for communication-intensive nested iterations, albeit at the cost of some extra overhead to construct and maintain the tree topology.

The algorithm presented in this paper can be viewed as a generalization¹ of [10] to the case of GEVCs rather than EVCs. The algorithm estimates the GEVCs corresponding to the Q largest or smallest GEVLs without explicitly constructing the two network-wide sensor signal covariance matrices that define them. Instead, the nodes exchange Q -dimensional compressed sensor observations which are used to construct local covariance matrices, followed by a local GEVD of these matrices. Due to the compression, the communication bandwidth is significantly reduced, and the local GEVDs are significantly cheaper to compute since the local covariance matrices have a much smaller dimension than the network-wide covariance matrices. For the sake of an easy exposition, we will only consider the case of a fully-connected WSN. However, it is noted that all results can be modified to the

The work of A. Bertrand was supported by a Postdoctoral Fellowship of the Research Foundation - Flanders (FWO). This work was carried out at the ESAT Laboratory of KU Leuven, in the frame of KU Leuven Research Council CoE PFV/10/002 (OPTEC), Concerted Research Action GOA-MaNet, the Belgian Programme on Interuniversity Attraction Poles initiated by the Belgian Federal Science Policy Office IUAP P7/23 (BESTCOM, 2012-2017), Research Projects FWO nr. G.0763.12 'Wireless acoustic sensor networks for extended auditory communication', FWO nr. G.0763.12 'Wireless acoustic sensor networks for extended auditory communication', FWO nr. G.0931.14 'Design of distributed signal processing algorithms and scalable hardware platforms for energy-vs-performance adaptive wireless acoustic sensor networks', and HANDiCAMS. The project HANDiCAMS acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 323944. The scientific responsibility is assumed by its authors.

¹For the algorithms in [8], [9], such a generalization is hampered by the use of Oja's learning rule or (sample-based) power iterations, which do not generalize to the computation of GEVCs.

case of multi-hop networks, using similar techniques as in [10].

II. PROBLEM STATEMENT

We consider a fully-connected WSN with a set of sensor nodes $\mathcal{K} = \{1, \dots, K\}$. Node k collects observations of the M_k -dimensional complex-valued stochastic sensor signals \mathbf{u}_k and \mathbf{v}_k , which are both (short-term) stationary and ergodic. We define $\mathbf{u}_k[t]$ as the t -th observation of \mathbf{u}_k , i.e., t denotes the sample index. Depending on the context, observations of \mathbf{u}_k and \mathbf{v}_k can be collected simultaneously (e.g., using two different sets of sensors per node) or sequentially. For example, the latter applies when the target signal switches between two states (e.g. ‘on’ or ‘off’ [4]), yielding a pair of covariance matrices (one for each state), on which a GEVD is computed. This approach is often used in speech enhancement applications [4] or applications with a controllable stimulus, such as evoked neuromagnetic experiments [3], [7].

We define the M -dimensional stochastic vectors \mathbf{u} and \mathbf{v} as the stacked version of all \mathbf{u}_k ’s and \mathbf{v}_k ’s, respectively, where $M = \sum_{k \in \mathcal{K}} M_k$. Without loss of generality (w.l.o.g.), we assume that \mathbf{u} and \mathbf{v} are zero-mean, possibly requiring a mean subtraction pre-processing step. The covariance matrices of \mathbf{u} and \mathbf{v} are then defined as

$$\mathbf{R}_{uu} = E\{\mathbf{u}\mathbf{u}^H\} \quad (1)$$

$$\mathbf{R}_{vv} = E\{\mathbf{v}\mathbf{v}^H\} \quad (2)$$

where $E\{\cdot\}$ denotes the expected value operator, and the superscript H denotes the conjugate transpose operator.

Let \mathbf{U} denote an $M \times N$ observation matrix containing N different observations of \mathbf{u} in its columns. Then ergodicity of \mathbf{u} implies that \mathbf{R}_{uu} can be approximated by the sample covariance matrix, i.e.,

$$\mathbf{R}_{uu} \approx \frac{1}{N} \mathbf{U}\mathbf{U}^H \quad (3)$$

and equality holds in the case of an infinite observation window, i.e., $\mathbf{R}_{uu} = \lim_{N \rightarrow \infty} \frac{1}{N} \mathbf{U}\mathbf{U}^H$. Similarly, $\mathbf{R}_{vv} \approx \frac{1}{N} \mathbf{V}\mathbf{V}^H$ where \mathbf{V} contains observations of \mathbf{v} in its columns.

A. Generalized eigenvalue decomposition (GEVD)

Computing the GEVD of the ordered matrix pair $(\mathbf{R}_{uu}, \mathbf{R}_{vv})$ consists of finding GEVCs \mathbf{x} and corresponding GEVLs λ such that

$$\mathbf{R}_{uu}\mathbf{x} = \lambda\mathbf{R}_{vv}\mathbf{x}. \quad (4)$$

There exist various techniques to compute the GEVCs and GEVLs defined in (4), for which we refer to [1]. Note that, if \mathbf{R}_{vv} has full rank, the GEVD can also be computed from the EVD of $\mathbf{R} = \mathbf{R}_{vv}^{-1}\mathbf{R}_{uu}$, although this is not recommended from a numerical point of view.

Let $\hat{\mathbf{X}}$ denote an $M \times Q$ matrix where the columns are the principal GEVCs corresponding to the Q largest GEVLs of $(\mathbf{R}_{uu}, \mathbf{R}_{vv})$. In the sequel, we assume w.l.o.g. that the GEVCs are always scaled such that $\hat{\mathbf{x}}^H \mathbf{R}_{vv} \hat{\mathbf{x}} = 1$ for any GEVC $\hat{\mathbf{x}}$. Since \mathbf{R}_{uu} and \mathbf{R}_{vv} are Hermitian and positive-definite, their GEVLs are real and their GEVCs are \mathbf{R}_{vv} -orthogonal, i.e.,

$\hat{\mathbf{X}}^H \mathbf{R}_{vv} \hat{\mathbf{X}} = \mathbf{I}_Q$ where \mathbf{I}_Q is the $Q \times Q$ identity matrix [1]. It can then be shown that $\hat{\mathbf{X}}$ is the solution of the constrained optimization problem [11]:

$$\hat{\mathbf{X}} = \arg \max_{\mathbf{X}} \text{Tr} \{ \mathbf{X}^H \mathbf{R}_{uu} \mathbf{X} \} \quad (5)$$

$$\text{s.t. } \mathbf{X}^H \mathbf{R}_{vv} \mathbf{X} = \mathbf{I}_Q \quad (6)$$

where $\text{Tr}\{\cdot\}$ denotes the trace operator. This also implies that the principal GEVC $\hat{\mathbf{x}}_1$ (the first column of $\hat{\mathbf{X}}$) maximizes the generalized Rayleigh coefficient [1]

$$\hat{\mathbf{x}}_1 = \arg \max_{\mathbf{x}} \frac{\mathbf{x}^H \mathbf{R}_{uu} \mathbf{x}}{\mathbf{x}^H \mathbf{R}_{vv} \mathbf{x}}. \quad (7)$$

Although we only focus on the principal GEVCs, it is noted that all results in this paper can straightforwardly be modified to also estimate the GEVCs corresponding to the Q *smallest* GEVLs. To do so, the max operator should be replaced by a min operator in (5) and (7), and in similar expressions in the sequel.

B. Applications of the GEVD

The interpretation of the principal GEVCs depends on the context in which they are used. For example, the GEVD is often used to compute spatial filters for noise reduction. Indeed, if \mathbf{u} and \mathbf{v} represent a target signal vector and a noise signal vector, respectively, then (7) implies that $\hat{\mathbf{x}}_1$ is the spatial filter that maximizes the SNR when applied to the signal $\mathbf{u} + \mathbf{v}$. In the general case, (5)-(6) implies that the columns of $\hat{\mathbf{X}}$ span the Q -dimensional subspace with maximal SNR. This can be viewed as a generalization of the well-known principal component analysis (PCA), which relies on an EVD to find the subspace with maximal variance. It is noted that $\hat{\mathbf{X}}$ also contains the max-SNR subspace if $\mathbf{u} = \mathbf{y} + \mathbf{v}$, i.e., a target signal \mathbf{y} contaminated with additive noise defined by the stochastic variable \mathbf{v} . This is exploited in applications where both ‘signal+noise’ and ‘noise-only’ samples are available [2]–[4]. In a classification context, in which observations of \mathbf{u} and \mathbf{v} have to be distinguished from each other, $\hat{\mathbf{X}}$ can be used to extract feature vectors with good discriminative properties [7]. If \mathbf{R}_{uu} and \mathbf{R}_{vv} represent two different correlation matrices computed from a single set of sensor signals (e.g., using two different time lags), $\hat{\mathbf{X}}$ will contain source separation filters [6].

C. Distributed vs. centralized computation

In the envisaged WSN, node k has access to M_k -dimensional observations of \mathbf{u}_k and \mathbf{v}_k , which represent only M_k rows of the observation matrices \mathbf{U} and \mathbf{V} . To estimate $\hat{\mathbf{X}}$, all nodes may transmit their observations to an FC, where \mathbf{R}_{uu} and \mathbf{R}_{vv} can be computed and updated at regular time intervals (e.g., based on (3) using the N most recent observations), followed by the computation of the Q principal GEVCs. However, transmitting the raw sensor signal observations requires a large communication bandwidth, and the computation of the GEVCs for large M requires significant computational power at the FC.

This paper proposes a distributed adaptive estimation algorithm where each node estimates a specific part of $\hat{\mathbf{X}}$,

avoiding the centralized computation of the full network-wide covariance matrices \mathbf{R}_{uu} and \mathbf{R}_{vv} . The algorithm is referred to as the distributed adaptive covariance-matrix generalized eigenvector estimation (DACGEE) algorithm.

III. DACGEE ALGORITHM

A. Preliminaries

The DACGEE algorithm iteratively updates the $M \times Q$ matrix \mathbf{X}^i , where i is the iteration index, with the goal of obtaining $\lim_{i \rightarrow \infty} \mathbf{X}^i = \hat{\mathbf{X}}$. We define the partitioning

$$\mathbf{X}^i = \begin{bmatrix} \mathbf{X}_1^i \\ \vdots \\ \mathbf{X}_K^i \end{bmatrix} \quad (8)$$

where \mathbf{X}_k^i is the part of \mathbf{X}^i that corresponds to node k and so to \mathbf{u}_k and \mathbf{v}_k , such that $\mathbf{X}^{iH} \mathbf{u} = \sum_{k \in \mathcal{K}} \mathbf{X}_k^{iH} \mathbf{u}_k$. Based on this partitioning, node k is responsible for updating the submatrix \mathbf{X}_k^i .

In between iteration i and iteration $i + 1$, node k also uses \mathbf{X}_k^i to compress its observations of \mathbf{u}_k and \mathbf{v}_k , hence \mathbf{X}_k^i serves both as an estimation variable and a compression matrix. The compressed versions of \mathbf{u}_k and \mathbf{v}_k are denoted as $\bar{\mathbf{u}}_k^i$ and $\bar{\mathbf{v}}_k^i$, respectively, and are computed as

$$\bar{\mathbf{u}}_k^i = \mathbf{X}_k^{iH} \mathbf{u}_k \quad (9)$$

$$\bar{\mathbf{v}}_k^i = \mathbf{X}_k^{iH} \mathbf{v}_k. \quad (10)$$

In between iteration i and iteration $i + 1$, each node compresses its new observations by means of (9)-(10) and broadcasts these compressed observations to the other nodes. For the sake of an easy exposition, we assume that $Q < M_k$, $\forall k \in \mathcal{K}$. If there exists a k for which $Q \geq M_k$, node k broadcasts uncompressed observations of \mathbf{u}_k and \mathbf{v}_k .

In the rest of this section, for the sake of conciseness, we only define the notation for variables referring to \mathbf{u} , and we implicitly assume a similar notation for \mathbf{v} .

Since the network is fully-connected, each node k collects observations of the $(M_k + (K - 1)Q)$ -dimensional signal

$$\tilde{\mathbf{u}}_k^i = \begin{bmatrix} \mathbf{u}_k \\ \bar{\mathbf{u}}_{-k}^i \end{bmatrix} \quad (11)$$

where $\bar{\mathbf{u}}_{-k}^i = [\bar{\mathbf{u}}_1^{iT} \dots \bar{\mathbf{u}}_{k-1}^{iT} \bar{\mathbf{u}}_{k+1}^{iT} \dots \bar{\mathbf{u}}_K^{iT}]^T$. We define the corresponding covariance matrix

$$\mathbf{R}_{\tilde{\mathbf{u}}_k^i}^i = E\{\tilde{\mathbf{u}}_k^i \tilde{\mathbf{u}}_k^{iH}\}. \quad (12)$$

For the sake of an easy notation, we define the matrix \mathbf{C}_k^i that allows to write $\tilde{\mathbf{u}}_k^i$ as a function of the full \mathbf{u} , i.e.,

$$\tilde{\mathbf{u}}_k^i = \mathbf{C}_k^{iH} \mathbf{u} \quad (13)$$

with

$$\mathbf{C}_k^i = \begin{bmatrix} \mathbf{O} & \mathbf{B}_{<k}^i & \mathbf{O} \\ \mathbf{I}_{M_k} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{B}_{>k}^i \end{bmatrix} \quad (14)$$

where \mathbf{O} is an all-zero matrix of appropriate dimension, and

with

$$\mathbf{B}_{<k}^i = \text{Blkdiag}(\mathbf{X}_1^i, \dots, \mathbf{X}_{k-1}^i) \quad (15)$$

$$\mathbf{B}_{>k}^i = \text{Blkdiag}(\mathbf{X}_{k+1}^i, \dots, \mathbf{X}_K^i) \quad (16)$$

where the operator $\text{Blkdiag}(\cdot)$ is used with a slight abuse of notation to denote a non-square block diagonal matrix built from the matrices in its argument. It is noted that

$$\mathbf{R}_{\tilde{\mathbf{u}}_k^i}^i = \mathbf{C}_k^{iH} \mathbf{R}_{uu} \mathbf{C}_k^i. \quad (17)$$

Similarly to (3), $\mathbf{R}_{\tilde{\mathbf{u}}_k^i}^i$ can be estimated at node k as

$$\mathbf{R}_{\tilde{\mathbf{u}}_k^i}^i \approx \frac{1}{N} \tilde{\mathbf{U}}_k \tilde{\mathbf{U}}_k^H \quad (18)$$

where $\tilde{\mathbf{U}}_k$ is an $(M_k + (K - 1)Q) \times N$ matrix, containing N observations of $\tilde{\mathbf{u}}_k^i$ in its columns, i.e., $\tilde{\mathbf{U}}_k = \mathbf{C}_k^{iH} \mathbf{U}$.

B. Algorithm derivation

We define the objective function

$$J(\mathbf{X}) = \text{Tr}\{\mathbf{X}^H \mathbf{R}_{uu} \mathbf{X}\}. \quad (19)$$

As mentioned earlier (see (5)), $\hat{\mathbf{X}}$ maximizes (19) under the constraint $\mathbf{X}^H \mathbf{R}_{vv} \mathbf{X} = \mathbf{I}_Q$. The starting point of the algorithm derivation is the following (centralized) alternating optimization (AO) procedure:

- 1) Set $i \leftarrow 0$, $q \leftarrow 1$, and \mathbf{X}^0 as a random $M \times Q$ matrix.
- 2) Choose \mathbf{X}^{i+1} as a solution of:

$$\max_{\mathbf{X}} J(\mathbf{X}) \quad (20)$$

$$\text{s.t. } \mathbf{X}^H \mathbf{R}_{vv} \mathbf{X} = \mathbf{I}_Q \quad (21)$$

$$\cdot \forall k \in \mathcal{K} \setminus \{q\} : \text{Range}\{\mathbf{X}_k\} = \text{Range}\{\mathbf{X}_k^i\} \quad (22)$$

where \mathbf{X}_k is the k -th submatrix of \mathbf{X} similarly defined as in (8), and where $\text{Range}\{\mathbf{X}_k^i\}$ denotes the subspace spanned by the columns of \mathbf{X}_k^i .

- 3) $i \leftarrow i + 1$ and $q \leftarrow (q \bmod K) + 1$.
- 4) Return to step 2.

In each iteration, the AO procedure can update one particular submatrix of \mathbf{X} freely (i.e., \mathbf{X}_q), while constraining the other submatrices to preserve their current column space. It is noted that the current point \mathbf{X}^i in iteration i is always in the constraint set such that $J(\mathbf{X}^{i+1}) \geq J(\mathbf{X}^i)$ (except at $i = 0$), i.e., by definition the objective function increases in a monotonic fashion. Since the AO procedure is defined in a centralized context², the addition of the constraints (22) seems somewhat artificial and unnecessary. However, the particular form of (22) is chosen such that this procedure can be executed in a distributed fashion, which is explained next.

The constraint (22) is equivalent to

$$\forall k \in \mathcal{K} \setminus \{q\}, \exists \mathbf{G}_k \in \mathbb{C}^{Q \times Q} : \mathbf{X}_k = \mathbf{X}_k^i \mathbf{G}_k \quad (26)$$

²Note that it requires the network-wide covariance matrices \mathbf{R}_{uu} and \mathbf{R}_{vv}

TABLE I
THE DACGEE ALGORITHM IN A FULLY-CONNECTED NETWORK

1) Set $i \leftarrow 0$, $q \leftarrow 1$, and initialize all \mathbf{X}_k^0 , $\forall k \in \mathcal{K}$, with random entries.
2) Each node $k \in \mathcal{K}$ broadcasts N new compressed observations $\bar{\mathbf{u}}_k^i[j] = \mathbf{X}_k^i H \mathbf{u}_k[iN + j]$ and $\bar{\mathbf{v}}_k^i[j] = \mathbf{X}_k^i H \mathbf{v}_k[iN + j]$ (where $j = 1 \dots N$).
3) At node q : <ul style="list-style-type: none"> • Estimate $\mathbf{R}_{\bar{\mathbf{u}}_q \bar{\mathbf{u}}_q}^i$ and $\mathbf{R}_{\bar{\mathbf{v}}_q \bar{\mathbf{v}}_q}^i$ similar to (18). • Compute the columns of $\tilde{\mathbf{X}}_q^{i+1}$ as the Q principal GEVCs of $(\mathbf{R}_{\bar{\mathbf{u}}_q \bar{\mathbf{u}}_q}^i, \mathbf{R}_{\bar{\mathbf{v}}_q \bar{\mathbf{v}}_q}^i)$. • Define $P = Q(K-1)$ and partition $\tilde{\mathbf{X}}_q^{i+1}$ as $\mathbf{X}_q^{i+1} = [\mathbf{I}_{M_k} \mathbf{O}] \tilde{\mathbf{X}}_q^{i+1} \quad (23)$ $\mathbf{G}_{-q} = [\mathbf{O} \mathbf{I}_P] \tilde{\mathbf{X}}_q^{i+1} \quad (24)$ and broadcast \mathbf{G}_{-q} to all the other nodes.
4) Each node $k \in \mathcal{K} \setminus \{q\}$ updates $\mathbf{X}_k^{i+1} = \mathbf{X}_k^i \mathbf{G}_k \quad (25)$
where $\mathbf{G}_{-q} = [\mathbf{G}_1^T \dots \mathbf{G}_{q-1}^T \mathbf{G}_{q+1}^T \dots \mathbf{G}_K^T]^T$.
5) $i \leftarrow i + 1$ and $q \leftarrow (q \bmod K) + 1$.
6) Return to step 2.

which allows to parameterize the optimization variable \mathbf{X} as

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1^i \mathbf{G}_1 \\ \vdots \\ \mathbf{X}_{q-1}^i \mathbf{G}_{q-1} \\ \mathbf{X}_q^i \\ \mathbf{X}_{q+1}^i \mathbf{G}_{q+1} \\ \vdots \\ \mathbf{X}_K^i \mathbf{G}_K \end{bmatrix}. \quad (27)$$

By stacking the optimization variables in $\tilde{\mathbf{X}} = [\mathbf{X}_1^T | \mathbf{G}_1^T | \dots | \mathbf{G}_{q-1}^T | \mathbf{G}_{q+1}^T | \dots | \mathbf{G}_K^T]^T$, and using the definition (14)-(16), we can rewrite (27) compactly as

$$\mathbf{X} = \mathbf{C}_q^i \tilde{\mathbf{X}}. \quad (28)$$

By using the parameterization (28) we can eliminate the constraint (22), and with (17), we find that solving (20)-(22) is equivalent to finding $\tilde{\mathbf{X}}_q^{i+1}$ as the solution of

$$\max_{\tilde{\mathbf{X}}} \text{Tr} \left\{ \tilde{\mathbf{X}}^H \mathbf{R}_{\bar{\mathbf{u}}_q \bar{\mathbf{u}}_q}^i \tilde{\mathbf{X}} \right\} \quad (29)$$

$$\text{s.t. } \tilde{\mathbf{X}}^H \mathbf{R}_{\bar{\mathbf{v}}_q \bar{\mathbf{v}}_q}^i \tilde{\mathbf{X}} = \mathbf{I}_Q \quad (30)$$

and setting $\mathbf{X}^{i+1} = \mathbf{C}_q^i \tilde{\mathbf{X}}_q^{i+1}$. Note that this optimization problem has the same form as (5)-(6), and can therefore be solved by performing a GEVD of $(\mathbf{R}_{\bar{\mathbf{u}}_q \bar{\mathbf{u}}_q}^i, \mathbf{R}_{\bar{\mathbf{v}}_q \bar{\mathbf{v}}_q}^i)$. Similar to (18), both $\mathbf{R}_{\bar{\mathbf{u}}_q \bar{\mathbf{u}}_q}^i$ and $\mathbf{R}_{\bar{\mathbf{v}}_q \bar{\mathbf{v}}_q}^i$ can be estimated at node q , and then their GEVD can be computed. The result can then be used to update the global \mathbf{X}^i into \mathbf{X}^{i+1} . The resulting DACGEE algorithm indeed exactly performs these operations, and is described in detail in Table I. As the DACGEE algorithm is then equivalent to the AO procedure, it will also result in a monotonic increase of $J(\mathbf{X})$ under the constraint (6).

Untuitively, since $\hat{\mathbf{X}}$ maximizes $J(\mathbf{X})$ under (6), we claim³ that \mathbf{X}^i indeed converges to $\hat{\mathbf{X}}$ under the DACGEE updates, which is empirically validated in Section IV.

Remark I: The DACGEE algorithm defined in Table I is assumed to operate in an adaptive, time-recursive context, where each iteration is performed over a different signal segment, i.e., the same block of samples is never transmitted more than once (this can also be inferred from the sample indices in step 2 of the algorithm). Note that the number of observations N that are collected and transmitted in between the iterations (step 2) should allow for a sufficiently accurate estimate of $\mathbf{R}_{\bar{\mathbf{u}}_q \bar{\mathbf{u}}_q}^i$ and $\mathbf{R}_{\bar{\mathbf{v}}_q \bar{\mathbf{v}}_q}^i$ in step 3. Note that, since $N \gg Q$, the additional communication cost to transmit the $P \times Q$ matrix \mathbf{G}_{-q} is negligible.

Remark II: It is noted that the DACGEE algorithm can be viewed as a generalization of the algorithm in [10], which computes the (non-generalized) EVCs of \mathbf{R}_{uu} in a similar fashion. Indeed, by setting $\mathbf{R}_{vv} = \mathbf{I}_M$ in (4) or (6), the GEVD problem becomes an EVD problem. Similar to [10], the DACGEE algorithm can also be extended to simply-connected networks (for further details, we refer to [10]).

Remark III: To measure sensor signal cross-correlations, the nodes continuously share compressed sensor signal observations (see step 2 in Table I). This requires more powerful -and hence more robust- communication links than what is traditionally envisaged in low-power WSNs where only parameters are exchanged between nodes. Furthermore, in GEVD-based signal enhancement applications [4], [6], the output of the algorithm consists of $\hat{\mathbf{X}}^H \mathbf{u}[t]$ and/or $\hat{\mathbf{X}}^H \mathbf{v}[t]$ for $t \in \mathbb{N}$, in which case the loss of a packet with samples would result in an immediate degradation of the output signal(s). Therefore, the envisaged WSNs are assumed to have robust communication links where link failures rarely occur. Nevertheless, due to its adaptive nature, the DACGEE algorithm is able to recover from link or node failures as long as these don't happen too frequently. More frequent link failures (without retransmission) will result in smaller sample sizes to populate the covariance matrices in step 3 of Table I, which may affect the estimation of $\hat{\mathbf{X}}$. This effect can be reduced by decreasing the updating frequency such that sufficient samples can be collected between updates, even under a substantial loss of packets (at the cost of a slower convergence). Permanent link failures can be handled by extending the DACGEE algorithm to simply-connected networks (see Remark II).

C. Communication cost and computational complexity

If $Q < M_k$, then the DACGEE algorithm reduces the communication cost for node k with a factor M_k/Q compared to the case where it would transmit its raw sensor signal observations to the other nodes or to an FC. Furthermore, in absence of a powerful FC, the inherent dimensionality reduction in the DACGEE algorithm also results in a reduced per-node computational complexity compared to the case where the centralized GEVD is computed in a single node. Indeed, assuming

³Note that the monotonic increase of $J(\mathbf{X})$ is not sufficient to prove convergence. However, an actual convergence proof can be constructed relying on a similar strategy as in [10] (details omitted, see also Remark II).

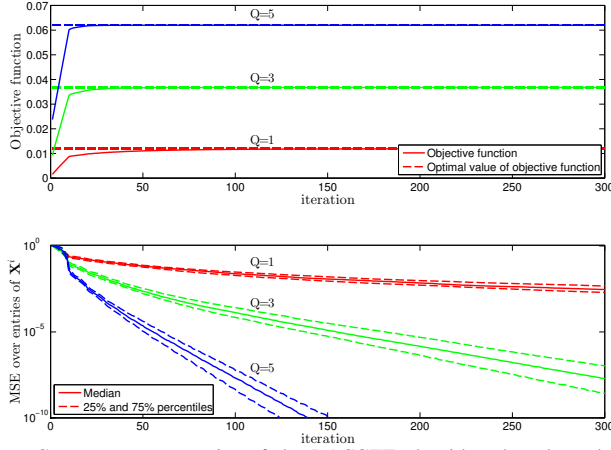


Fig. 1. Convergence properties of the DACGEE algorithm, based on the evaluation of the objective function (19) and the MSE over the entries of \mathbf{X}^i .

that the algorithm operates in an adaptive context where the GEVCs are continuously updated, a centralized GEVD has a complexity of $O(M^3)$ per update, whereas the DACGEE algorithm has a complexity of $O((M_q + (K-1)Q)^3)$ at the updating node q . For example, if $K = 10$, $Q = 1$, and $M_k = 15$, $\forall k \in \mathcal{K}$, then the centralized GEVD requires $\sim 3.4 \times 10^6$ flops to (re-)estimate \mathbf{X}^i , whereas the DACGEE algorithm requires only $\sim 1.3 \times 10^4$ flops per update. Of course, this comes with the drawback of having a slower adaptation speed or tracking performance due to the iterative nature of the DACGEE algorithm.

Remark IV: A tracking vs. communication/complexity trade-off can be obtained when computing multiple iterations of the DACGEE algorithm on the same block of N observations. Indeed, this would improve the adaptation speed and accuracy, but at the price of an increased communication cost since the same block of observations has to be transmitted multiple times. However, note that only the updating node has to retransmit its compressed observations, since the compressors at the other nodes have not changed (the $Q \times Q$ transformation of the compressors in (25) merely results in a $Q \times Q$ transformation of the previously compressed/transmitted observations).

IV. SIMULATIONS

In this section, we provide Monte-Carlo (MC) simulations⁴ of the DACGEE algorithm, and compare it with a centralized algorithm operated in an FC. In each MC run, a new scenario is created with $K = 10$ nodes, each collecting observations of a different 15-dimensional stochastic sensor signal \mathbf{u}_k , $\forall k \in \mathcal{K}$. This vector consists of a mixture of 10 spatially located source signals, from which Q are selected as target signals (having an on-off behavior), and the other $10 - Q$ are interfering signals (which are continuously active). During activity of the Q target sources, observations of the stacked vector \mathbf{u} are generated as

$$\mathbf{u}[t] = \mathbf{A}_{\text{on}} \cdot \mathbf{d}[t] + \mathbf{n}[t] \quad (31)$$

where \mathbf{A}_{on} is a deterministic $15K \times 10$ matrix (independent from t) for which in each MC run the entries are randomly drawn from a uniform distribution over the interval $[-0.5; 0.5]$, $\mathbf{d}[t]$ is an observation of a 10-dimensional stochastic signal from which the entries are independent and uniformly distributed over the interval $[-0.5; 0.5]$ and $\mathbf{n}[t]$ is an observation of a $15K$ -dimensional stochastic signal from which the entries are independent and uniformly distributed over the interval $[-\sqrt{0.1}/2; \sqrt{0.1}/2]$ (modelling spatially uncorrelated sensor noise). During inactivity of the Q target sources, the nodes collect ‘noise-only’ observations which are stacked in \mathbf{v} , and which are generated as

$$\mathbf{v}[t] = \mathbf{A}_{\text{off}} \cdot \mathbf{d}[t] + \mathbf{n}[t] \quad (32)$$

where $\mathbf{d}[t]$ and $\mathbf{n}[t]$ are generated by the same stochastic process as in (31), and \mathbf{A}_{off} is equal to \mathbf{A}_{on} , except for the first Q columns which are set to zero, indicating that the Q target sources are not active.

The upper part of Fig. 1 shows the monotonic increase of the objective function (19) over the different iterations of the DACGEE algorithm for different values of Q (averaged over 200 MC runs). We observe that, after a sufficient number of iterations, the algorithm always converges to the correct value. The bottom part of Fig. 1 shows the 25%, 50%, and 75% percentile (over 200 MC runs) of the squared error between \mathbf{X}^i and $\hat{\mathbf{X}}$, averaged over the MQ entries. It is observed that a larger Q yields a faster convergence, which is due to the extra degrees of freedom in the constraint (22).

V. CONCLUSIONS

We have described a distributed algorithm, referred to as the DACGEE algorithm, to estimate the principal GEVCs of a pair of sensor signal covariance matrices in a fully-connected WSN. The algorithm computes these GEVCs in an iterative fashion without explicitly constructing the network-wide covariance matrices. Instead of transmitting the raw sensor signal observations, the nodes only exchange compressed observations, providing a significant reduction in communication and computational costs, compared to a centralized approach. The algorithm has been validated by means of numerical simulations. Finally, it is noted that the algorithm can be modified to multi-hop networks, using similar techniques as in [10].

REFERENCES

- [1] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed. Baltimore: The Johns Hopkins University Press, 1996.
- [2] R. Nadakuditi and J. Silverstein, “Fundamental limit of sample generalized eigenvalue based detection of signals in noise using relatively few signal-bearing and noise-only samples,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 3, pp. 468–480, 2010.
- [3] K. Sekihara, D. Poeppel, A. Marantz, H. Koizumi, and Y. Miyashita, “MEG spatio-temporal analysis using a covariance matrix calculated from nonaveraged multiple-epoch data,” *IEEE Transactions on Biomedical Engineering*, vol. 46, no. 5, pp. 515–521, 1999.
- [4] S. Doclo and M. Moonen, “GSVD-based optimal filtering for single and multimicrophone speech enhancement,” *IEEE Transactions on Signal Processing*, vol. 50, no. 9, pp. 2230 – 2244, Sep. 2002.
- [5] N. Ito, E. Vincent, N. Ono, and S. Sagayama, “Robust estimation of directions-of-arrival in diffuse noise based on matrix-space sparsity,” INRIA, Technical report RR-8120, Oct. 2012. [Online]. Available: <http://hal.inria.fr/hal-00746271>

⁴The Matlab code that was used in these simulations can be downloaded from <http://homes.esat.kuleuven.be/~abertran/software.html>

- [6] A. M. Tomé, "The generalized eigendecomposition approach to the blind source separation problem," *Digital Signal Processing*, vol. 16, no. 3, pp. 288 – 302, 2006.
- [7] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K.-R. Muller, "Optimizing spatial filters for robust EEG single-trial analysis," *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 41–56, 2008.
- [8] A. Scaglione, R. Pagliari, and H. Krim, "The decentralized estimation of the sample covariance," in *Asilomar Conference on Signals, Systems and Computers*, oct. 2008, pp. 1722 –1726.
- [9] L. Li, A. Scaglione, and J. Manton, "Distributed principal subspace estimation in wireless sensor networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 725–738, 2011.
- [10] A. Bertrand and M. Moonen, "Distributed adaptive estimation of covariance matrix eigenvectors in wireless sensor networks with application to distributed PCA," *Signal Processing*, vol. 104, pp. 120–135, 2014.
- [11] K. Kokiopoulou, J. Chen, and Y. Saad, "Trace optimization and eigenproblems in dimension reduction methods," *Numerical Linear Algebra with Applications*, vol. 18, no. 3, pp. 565–602, May 2011.